

HPC-Projekte in der Luft- und Raumfahrt

Dr. Margrit Klitz

German Aerospace Center (DLR)

Simulation and Software Technology

Department High-Performance Computing

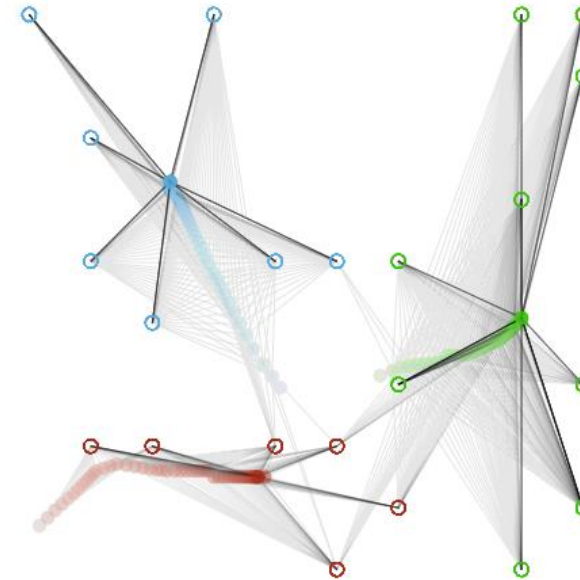
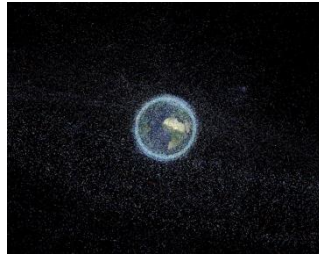
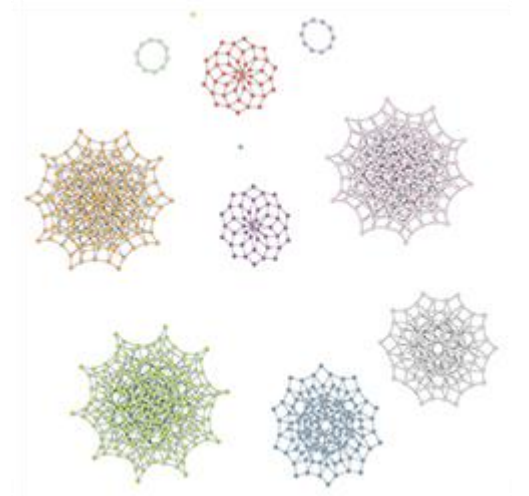
Linder Höhe, Cologne, Germany

A large, curved image of the Earth from space occupies the bottom right portion of the slide. It shows a view of the Earth's surface with blue oceans, green landmasses, and white clouds. The curvature of the planet is clearly visible, and the image is positioned as if looking down from a high altitude or from space.

Knowledge for Tomorrow

Inhalt

- Unsere Abteilung **SC-HPC im DLR**
- **Parallele Coupling Frameworks**
 - Für Flugzeugsimulation
 - Für Hubschraubersimulation
 - Mit Parallelisierung für GPUs
- **Weltraumschrott**
- **Essex goes Oakforest-PACS**
- **Paralleles Maschinelles Lernen**
 - Softwareframework HeAT



Das DLR

Deutsches Zentrum für Luft- und Raumfahrt



- Forschungseinrichtung
 - Forschungs- und Entwicklungsarbeiten in Luftfahrt, Raumfahrt, Energie, Verkehr, Digitalisierung und Sicherheit
 - nationale und internationale Kooperationen
- Raumfahrtagentur
 - Planung und Umsetzung der deutschen Raumfahrtaktivitäten
- Projektträger
 - Forschungsförderung



DLR Locations and Employees

Approx. 8000 employees across
40 institutes and facilities at 20 sites.

Offices in Brussels, Paris,
Tokyo and Washington.



Simulations- und Softwaretechnik

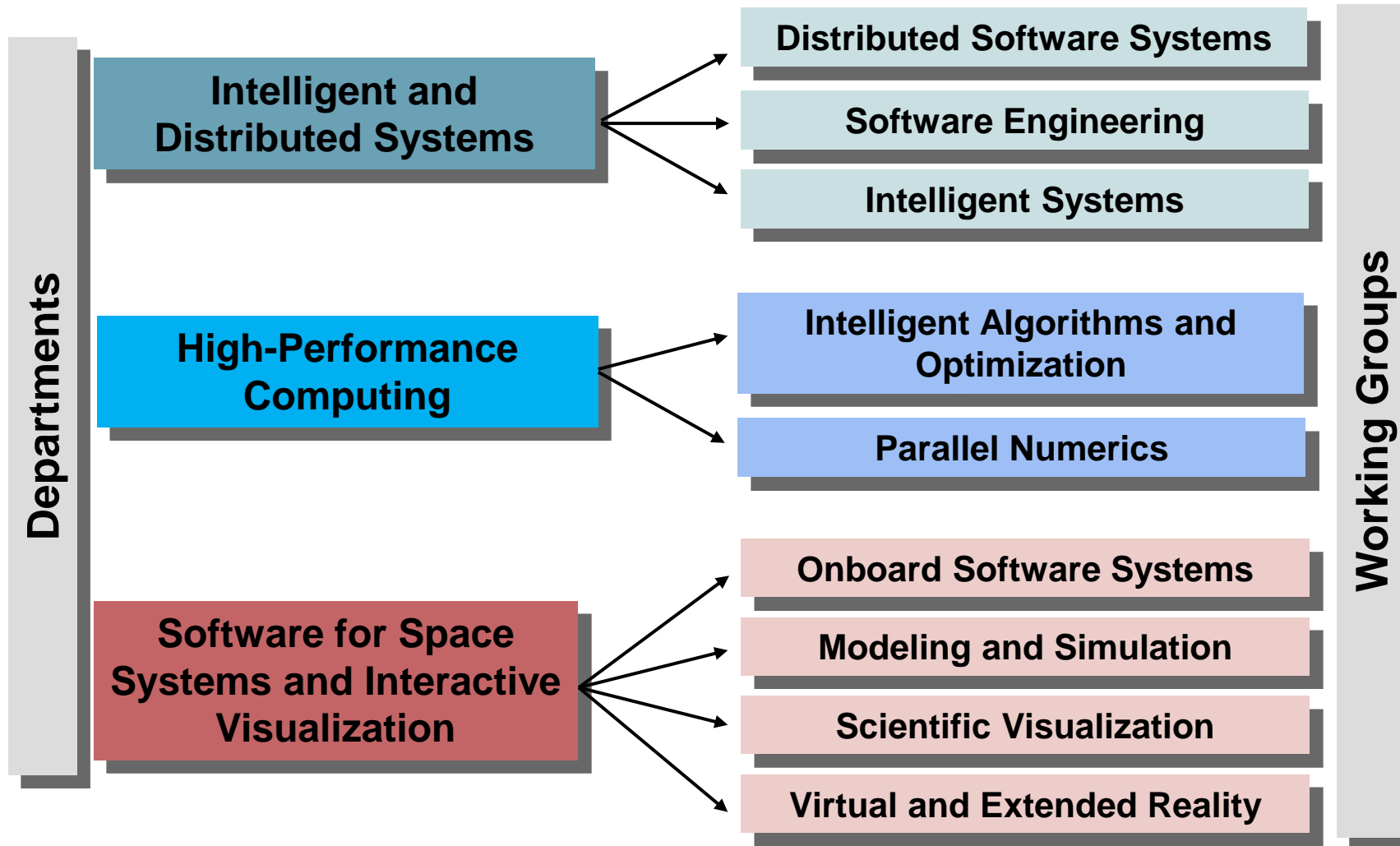


- Steht für **innovatives Software Engineering**,
- Entwickelt **anspruchsvolle Individualsoftwarelösungen** für das DLR und
- Ist Partner in **wissenschaftlichen Projekten** im Bereich Simulations- und Softwaretechnologie



DLR Institute Simulation and Software Technology

Scientific Themes and Working Groups



High Performance Computing Teams



**Department
High Performance Computing**
Head: Dr. Achim Basermann
Deputy: Dr. Margrit Klitz

**Intelligent Algorithms
& Optimization**
Dr. Martin Siggel

Quantum Computing

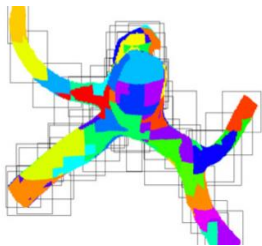
Parallel Numerics
Dr. Jonas Thies



High Performance Computing – Themen

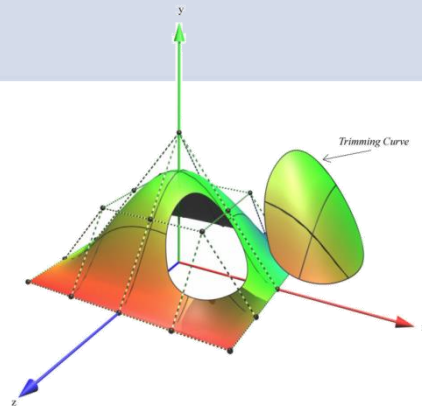
Parallele Numerische Mathematik

- Lösung großer linearer Gleichungssysteme und Eigenwertprobleme
- Deformation numerischer Gitter
- Adaptive Gitterverfeinerung / Gitter mit hängenden Knoten
- Simulation von Mehrkörpersystemen



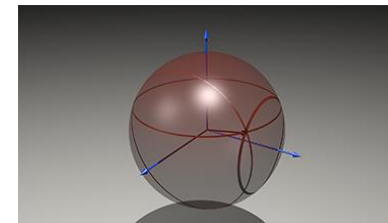
Modellierung und Formoptimierung

- Geometriemodellierung
- Multidisziplinäre Designoptimierung
- Automatisierte Gebietszerlegung



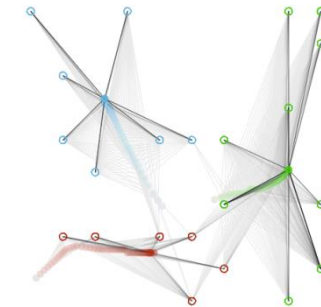
Software für zukünftige Rechnerarchitekturen

- Parallele und hybrid parallele Programmierung
- Parallele Bibliotheken für Graphik- oder Vielkernprozessoren
- Algorithmen für Quantencomputer



Effiziente Behandlung großer Datenmengen und Big Data

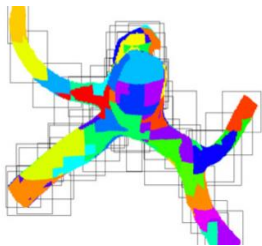
- Maschinelles Lernen
- Numerische und statistische Datenanalyse
- Verteilte Datenanalyse für Raumfahrtanwendungen
- Datenmanagement von HPC-Simulationen



High Performance Computing – Themen

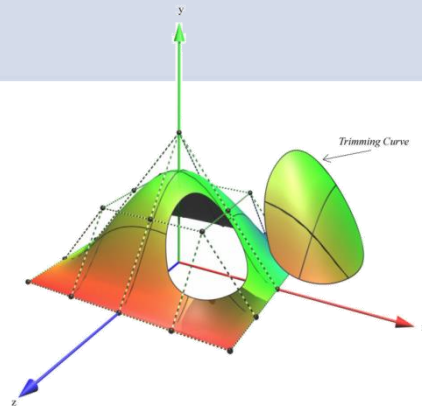
Parallele Numerische Mathematik

- Lösung großer linearer Gleichungssysteme und Eigenwertprobleme
- **Deformation numerischer Gitter**
- Adaptive Gitterverfeinerung / Gitter mit hängenden Knoten
- Simulation von Mehrkörpersystemen



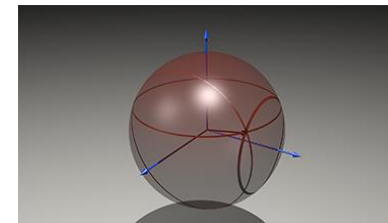
Modellierung und Formoptimierung

- **Geometriemodellierung**
- **Multidisziplinäre Designoptimierung**
- **Automatisierte Gebietszerlegung**



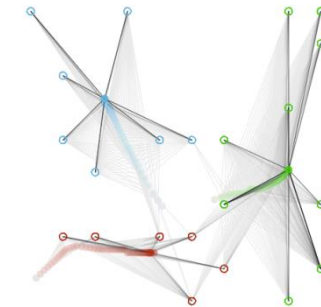
Software für zukünftige Rechnerarchitekturen

- Parallele und hybrid parallele Programmierung
- Parallele Bibliotheken für Graphik- oder Vielkernprozessoren
- **Algorithmen für Quantencomputer**

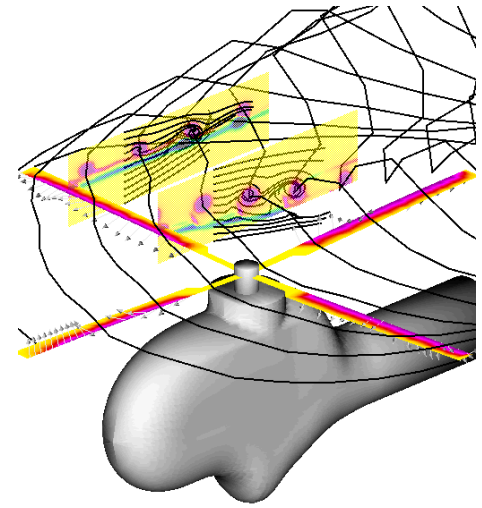
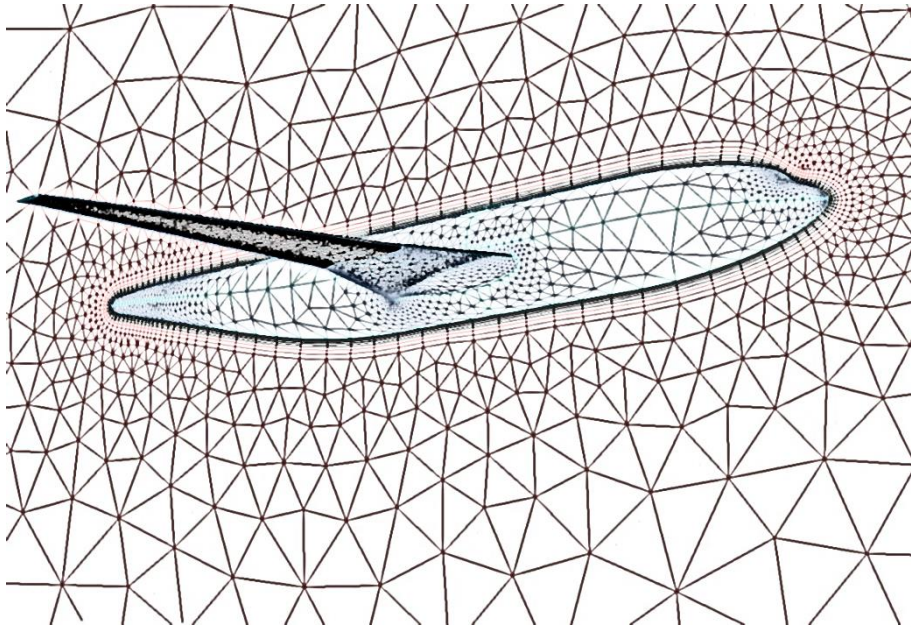


Effiziente Behandlung großer Datenmengen und Big Data

- Maschinelles Lernen
- Numerische und statistische Datenanalyse
- Verteilte Datenanalyse für Raumfahrtanwendungen
- Datenmanagement von HPC-Simulationen

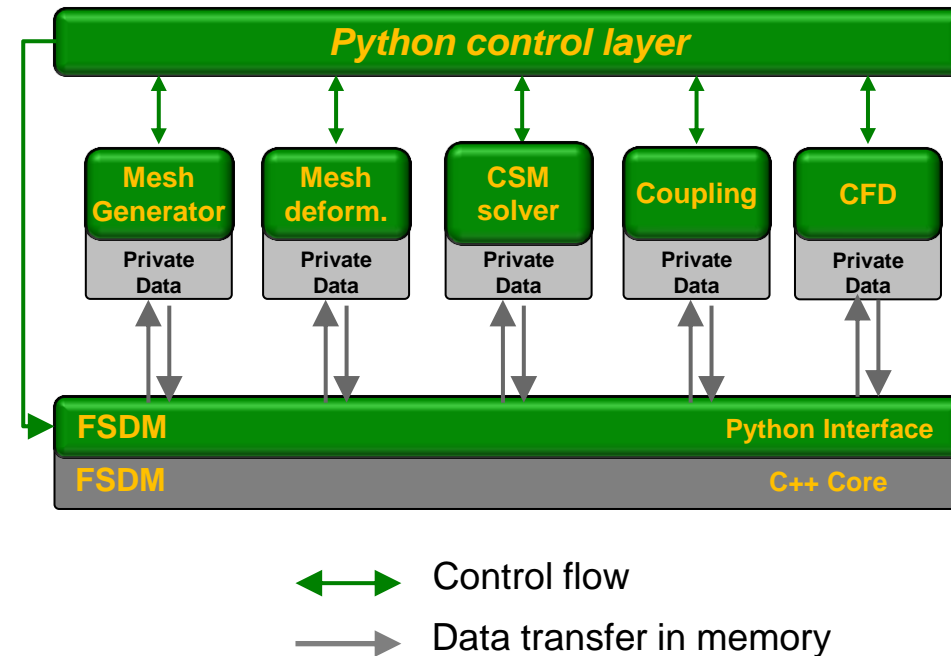


Parallel Coupling Frameworks



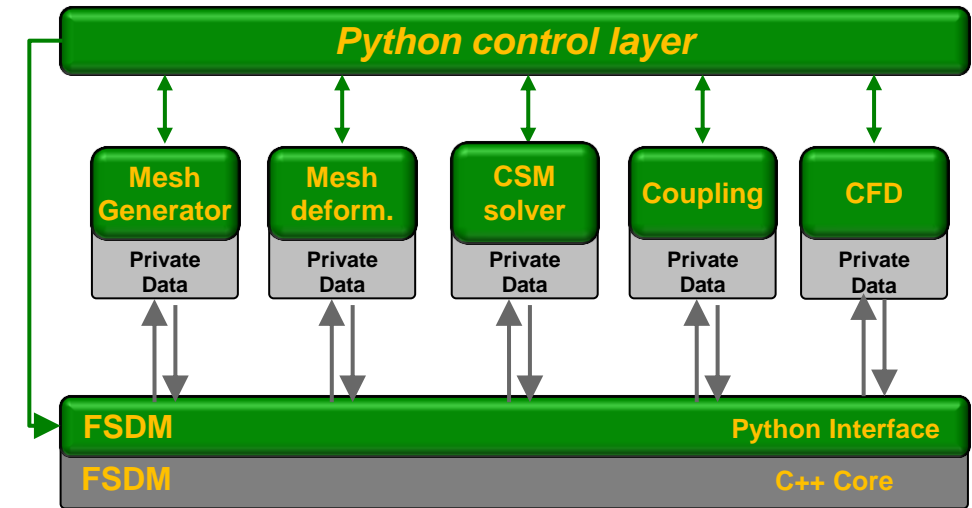
FlowSimulator

- HPC environment for integration of multiple parallel components into a process chain
- Jointly developed by Airbus, DLR, ONERA, universities, ...
- Components of simulation process chain („Plug-ins“) integrated via
 - Python control interface
 - FSDM data interface



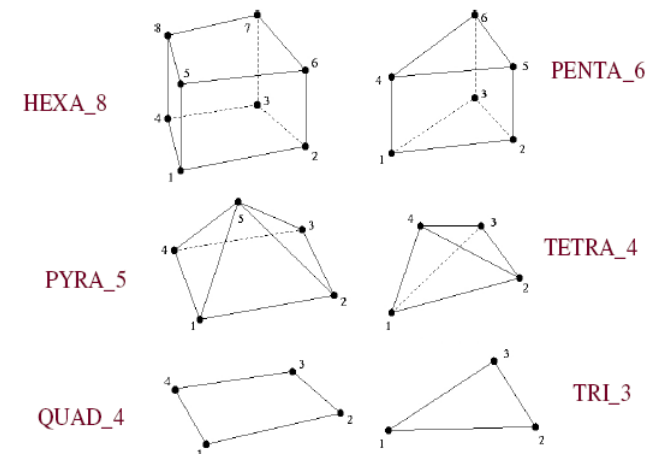
FlowSimulator DataManager (FSDM)

- FSDM reads/writes data (mesh, solution, log-data) from/to files
- FSDM decomposes data and distributes it over the different MPI domains
- FSDM stores data in container classes (e.g. FSMesh, FSDataSet)
- FSDM offers an interface (Python and C++) to container classes
- FSDM for us means unstructured meshes, can handle structured meshes as well
- FSDM is Open Source



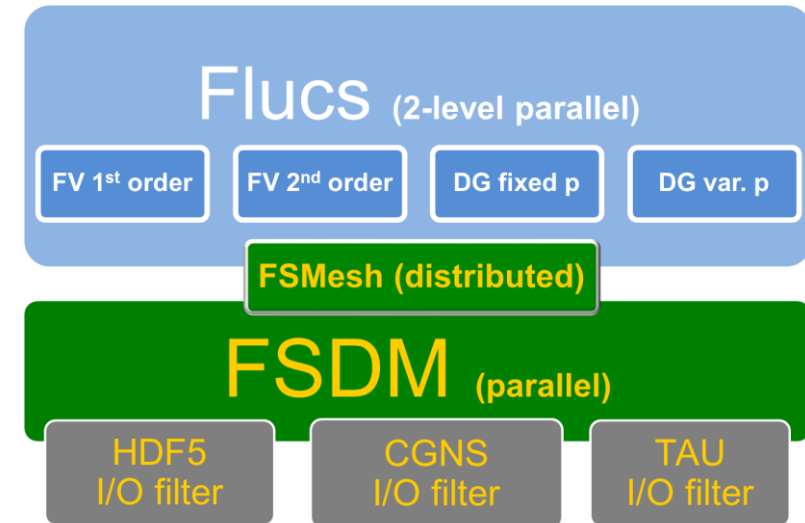
↔ Control flow

→ Data transfer in memory



FLUCS the **F**lexible **U**nstructured **C**FD **S**oftware

- The “next generation” flow solver currently developed at DLR by Institute for Aerodynamics and Flow Technology
- Solves the Euler-equations, the Navier-Stokes equations, or the RANS equations
- Two discretizations
 - Second-order Finite-Volume
 - Discontinuous Galerkin
- Flucs is designed as an FS plug-in in order facilitate multi-disciplinary simulations



➡ Consequently, development of FSDM and Flucs has to go hand in hand



Current research activity: Hanging Nodes/Edges/Faces

One of the main aims of the DLR: Virtual design of an aircraft.

- Determine flight characteristics by numerical simulation
 - Key element: numerical flow simulation
 - ➡ CFD software Flucs

Complex 3d transient flows

- highly time-consuming
 - ➡ use mesh adaptivity

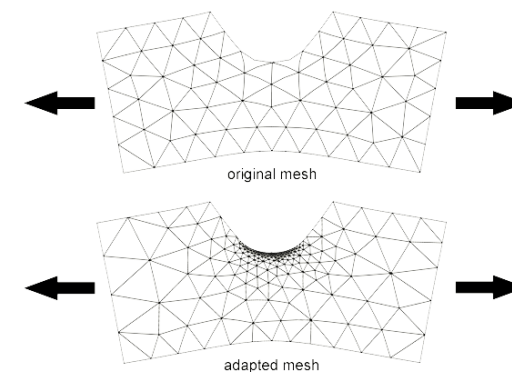
Ok, we are working on a tool for mesh adaptation

- ✗ Creation of hanging mesh entities and non-conforming interfaces

- Multidisciplinary optimization of a transport aircraft configuration.

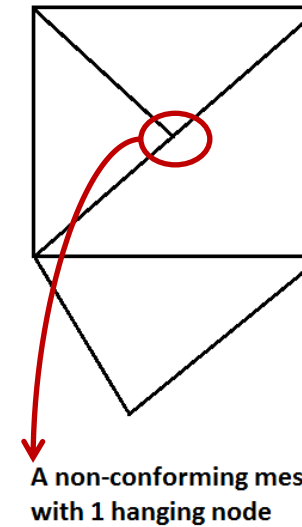


- Example of mesh adaption from Abaqus User's manual

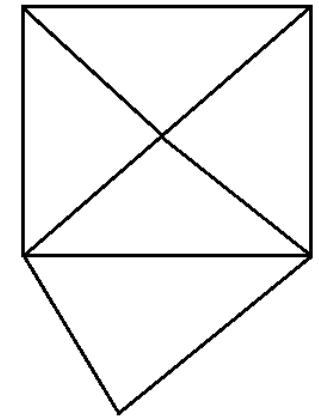


Motivation for Allowing Hanging Nodes/Edges/Faces

- Adaptation leads to **creation of hanging nodes** along non-conforming interfaces
 - ✗ E.g. disturb continuity of finite element space
 - ✗ Much effort required to remove them
- **But for Flucs:**
 - ✓ Discontinuous Galerkin & Finite Volume methods allow for very general non-matching grids with hanging nodes
 - ✓ Allow for very flexible grid structures and flexible adaptivity



A non-conforming mesh
with 1 hanging node

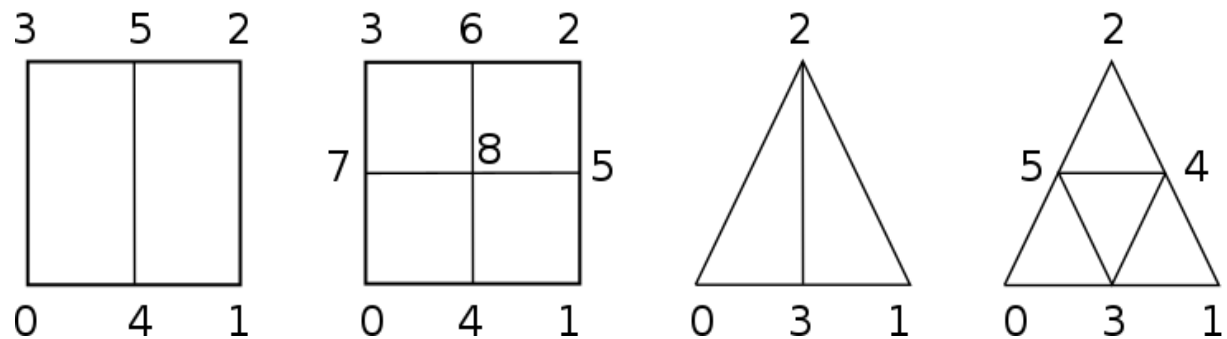
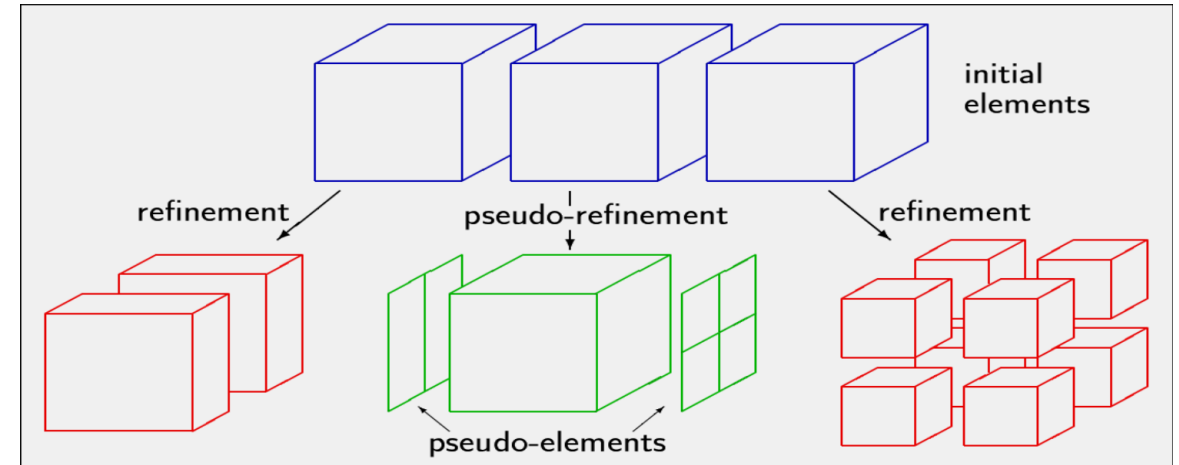


A conforming mesh

→ So far not accounted for in FSDM

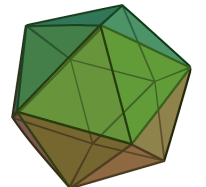
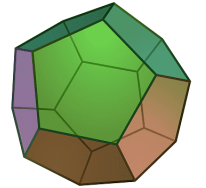
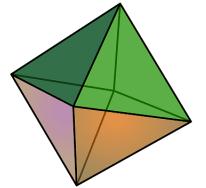
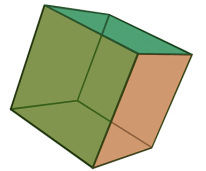
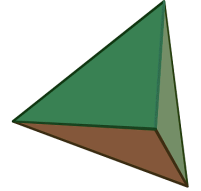
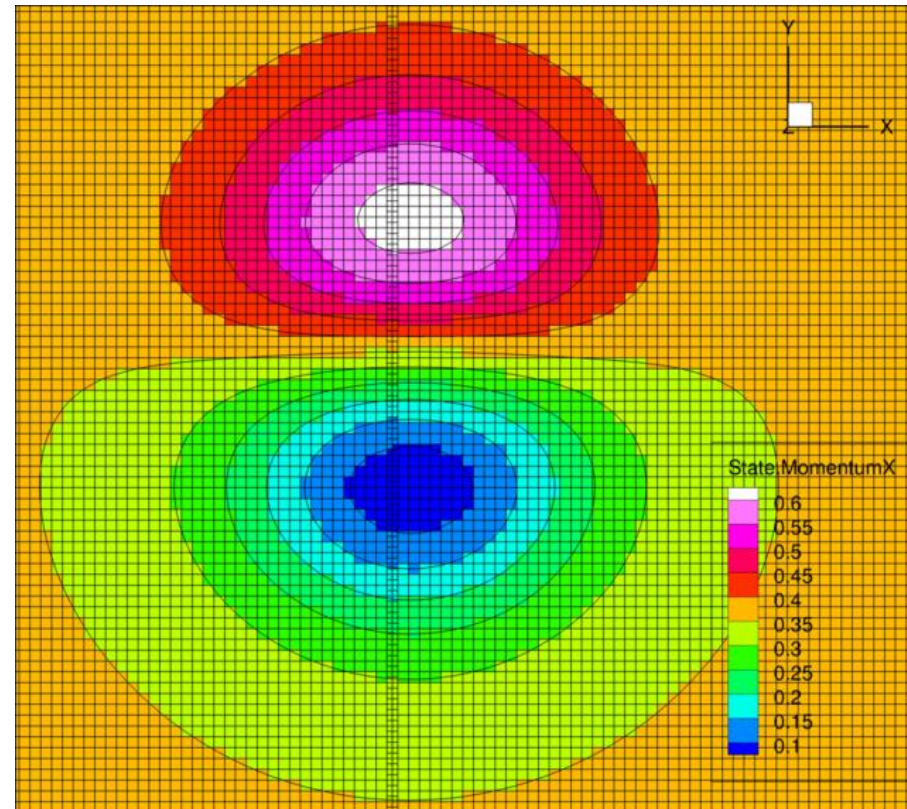
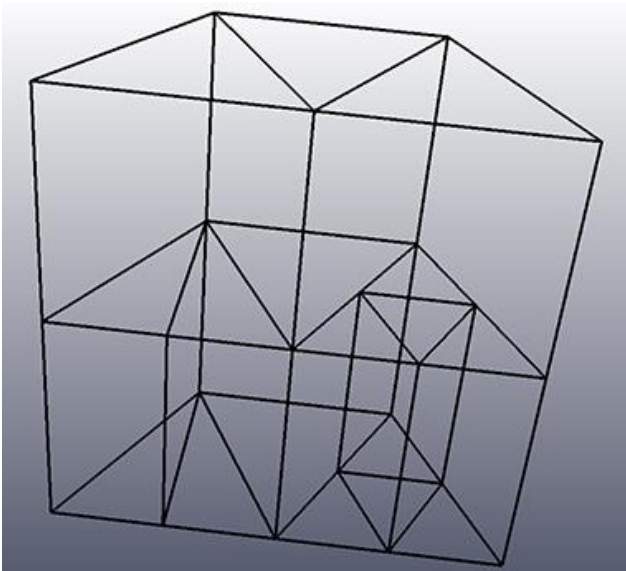
Basic Idea for Hanging Edges

- Pseudo element types represent the types of “hanging” connections, one possibility of storing the hanging connectivity,
- Ignored by the solver, have no volume and no solution values, used by the face extractor to create the face based grid,
- Complete the hanging node grid to a kind of (pseudo) conformity,
- Enable the adaptation to work on hanging elements in the same way as on conforming elements.
- Only four types of pseudo elements needed

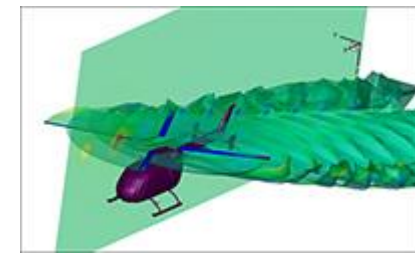


Wrapping it up

- FSDM now knows how to deal with Pseudo Cells
- Tricky part: Parallel face extraction
- Test with Flucs: Good results
- Go well with adaptivity
- Next: General polyhedric cells in FSDM

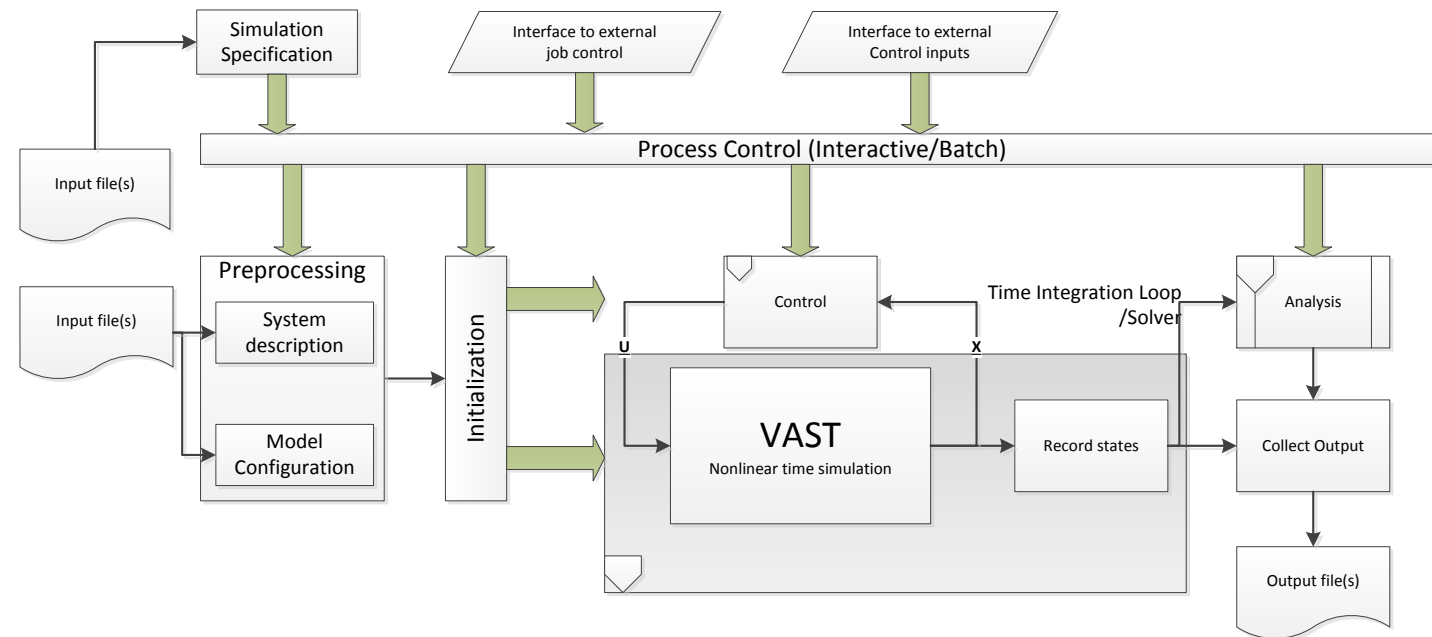


Original: Kjell André
Vector: DTR

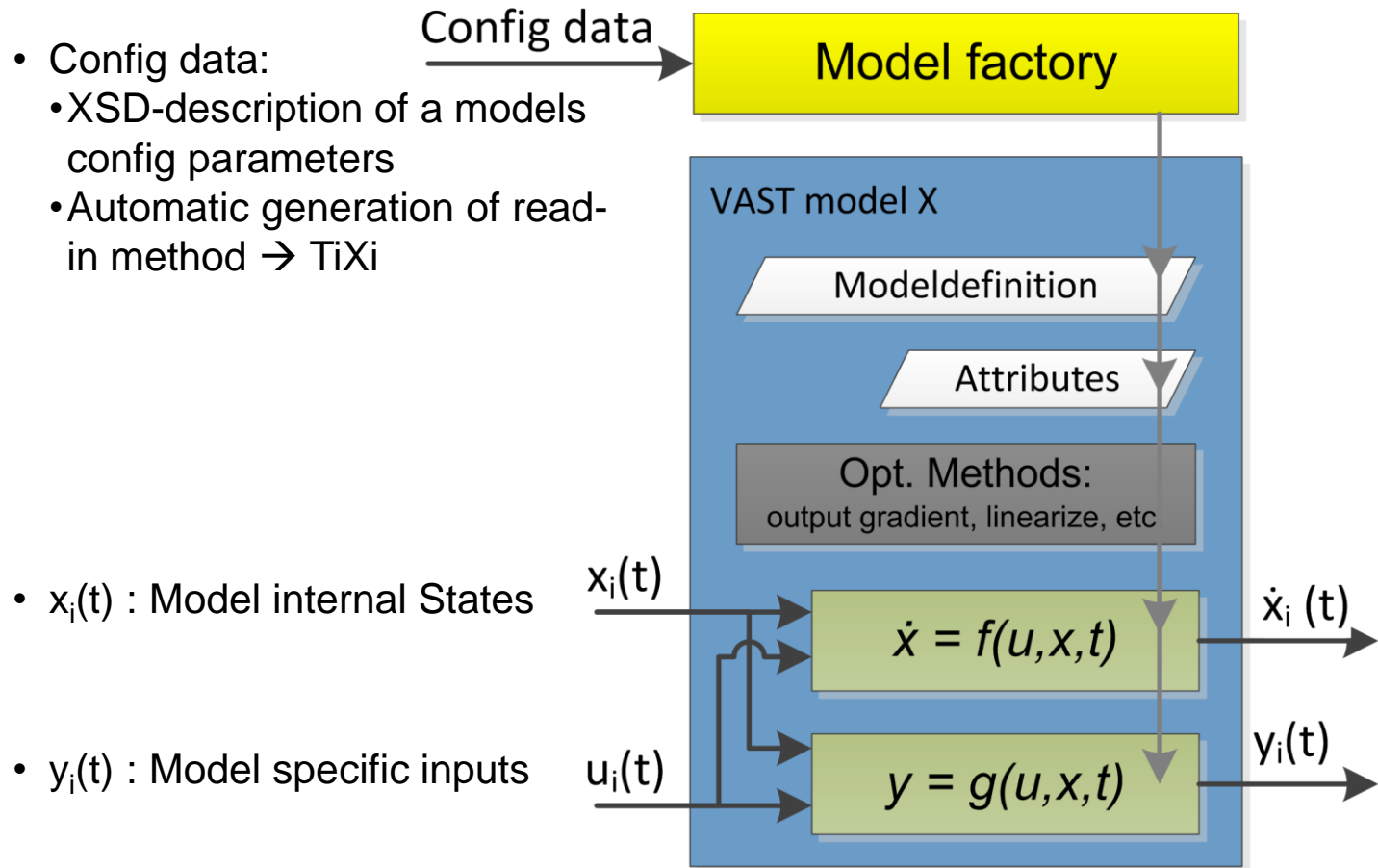


Versatile Aeromechanic Simulation Tool (VAST)

- Multiphysik- und Multimodellsimulationswerkzeug zur Beschreibung der Aeromechanik von Helikoptern
- **Ziel:** Beschreiben des dynamische Verhaltens, z.B. für die Beschreibung des freifliegenden Hubschraubers.
- **Wenig spezialisiert:** Framework, numerische Methoden, Benutzerschnittstellen für allgemeine Multimodellsimulationen ausgelegt
- **Modellierung** als gekoppelte ODEs: Unabhängige Entwicklung von Modellen für z.B. starre Strukturen (MBS), flexible Balken, Blattaerodynamik ...



VAST – Generic State Space Model

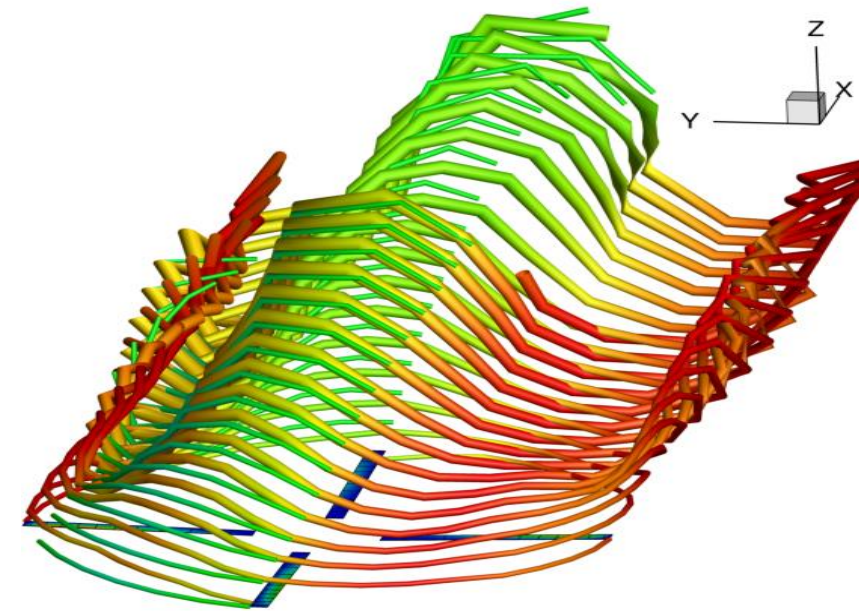
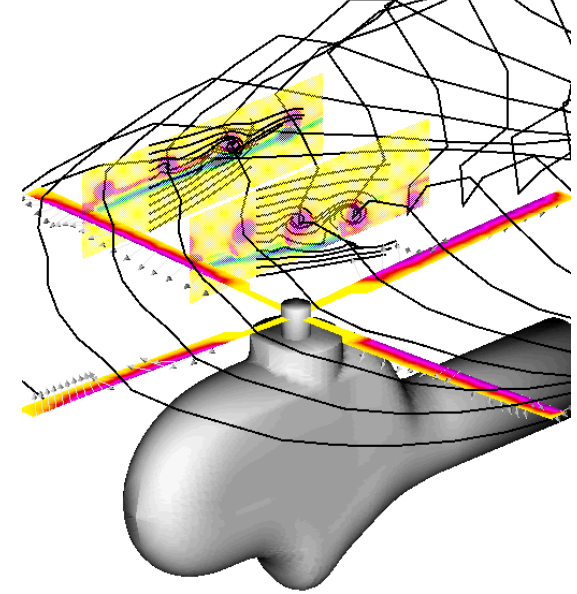


- Model factory: sets up and configures the model
- Modeldefinition: defines the model to the outside:
 - States, inputs, outputs
- Attributes: Model parameters processed by the factory needed for computation
- Optional Methods: Gradients and linearization can be provided, otherwise they are computed by the solver when needed
- Simulation Methods:
 - $\dot{x} = f(u, x, t)$: Compute first time derivatives of the states
 - $y = g(u, x, t)$: Compute outputs



DLR Software Free-Wake

- Entwickelt 1994-1996 vom Institut für Flugsystemtechnik
 - Fortran
 - MPI-parallel
- Wird vom FT-Rotor-Simulationscode S4 verwendet
- Simuliert die Strömung um den Rotor eines Hubschraubers
 - Vortex-Gitter-Methode
 - Diskretisiert komplexe Wirbelstrukturen mit einer Reihe von Wirbelelementen
 - Basierend auf experimentellen Daten (aus dem internationalen HART-Programm 1995)
- **Unsere Aufgabe:** hybride Free-Wake-Parallelisierung für CPUs und GPGPUs



OpenACC: Überblick

Vorteile:

- Spracherweiterungen ähnlich zu OpenMP
- API beschreibt eine Sammlung von Compiler-Direktiven, um Schleifen und Codebereiche von der HOST-CPU in angeschlossenen Beschleuniger auszulagern
→ Automatische Parallelisierung von seriellem Sourcecode

Nachteile

- Explizite Datenbewegung zwischen Host und GPU (bottleneck!)
- Automatische Parallelisierung nicht immer optimale Lösung

Unterstützte Sprachen:

- C
- C++
- **Fortran**

Unterstützte Compiler:

- CAPS
- CRAY
- **PGI**
- GCC



GPU Computing with OpenACC

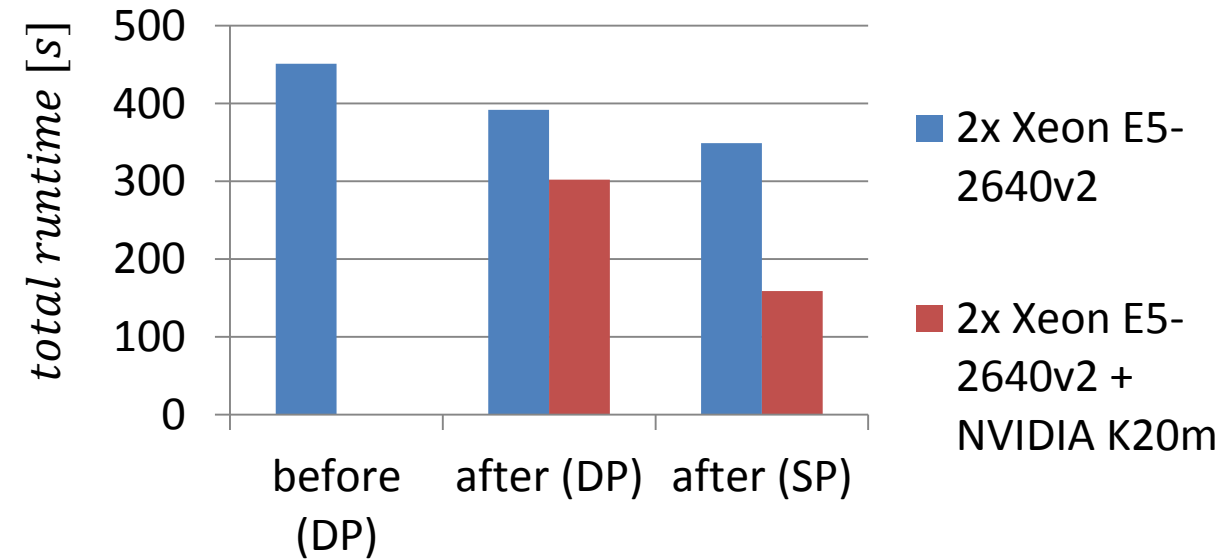
```
program main
  real :: a(N)
  ...
  !$acc data copyout(a)
    ! computation on the GPU in several loops:
    ...
    !$acc parallel loop
      do i = 1, N
        a(i) = 2.5 * i
      end do
    !$acc parallel loop
    ...
  !$acc end data
    ! Now results available on the CPU
  ...
end program main
```

Allokiert Speicher auf der GPU und kopiert
beim Verlassen des Bereichs die Daten zum Host.



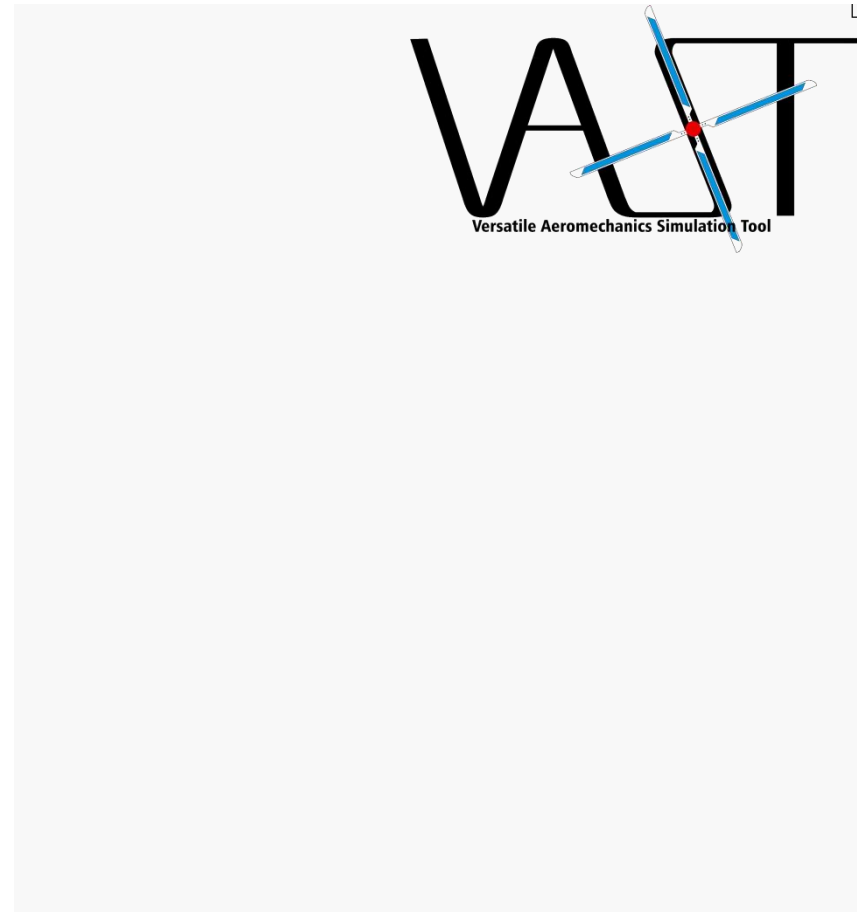
Port to GPGPU and Modernization

- Successfully ported Free-Wake simulation to GPUs using OpenACC
 - original numerical method not modified
 - results verified in SP and DP on CPU and GPU
 - “time to solution” improved significantly (for SP and CPU+GPU)
- Porting complex algorithms to GPUs is difficult
 - branches in loops hurt (much more than for CPUs)
- Loop restructuring may also improve the CPU performance
 - SIMD vectorization on modern CPUs
- Stumbled upon several OpenACC PGI-compiler bugs (all fixed by now)
- Freewake on a workstation with reasonable cycle times → Goal achieved!



VAST + Free-Wake

“Trim” of a free-flying, multi-rotor helicopter with rigid blades



Weltraumschrott



Picture from
<http://kidsnews.hu/2018/03/az-urszemetrol/>

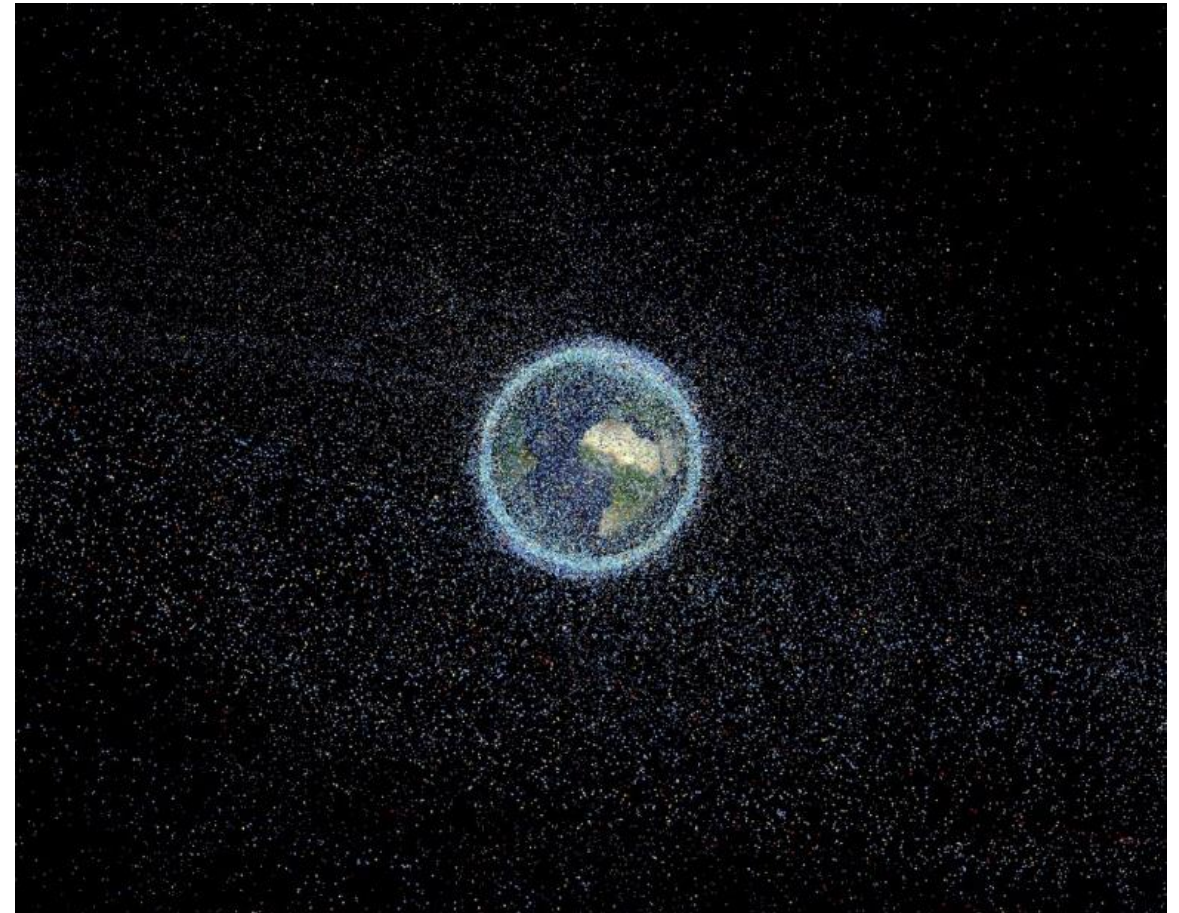
Weltraumschrott

Bezeichnet alle nicht-aktiven, nicht-kooperativen orbitalen Objekte, wie z.B.

- Ausgebrannte oder defekte Satelliten
- Verlorene Werkzeuge
- Trümmer aller Art (z.B. aus Kollisionen von Satelliten)
- Gefahr bereits ab 1 cm Größe
- 1 000 000 Objekte, rund 16.000 katalogisiert

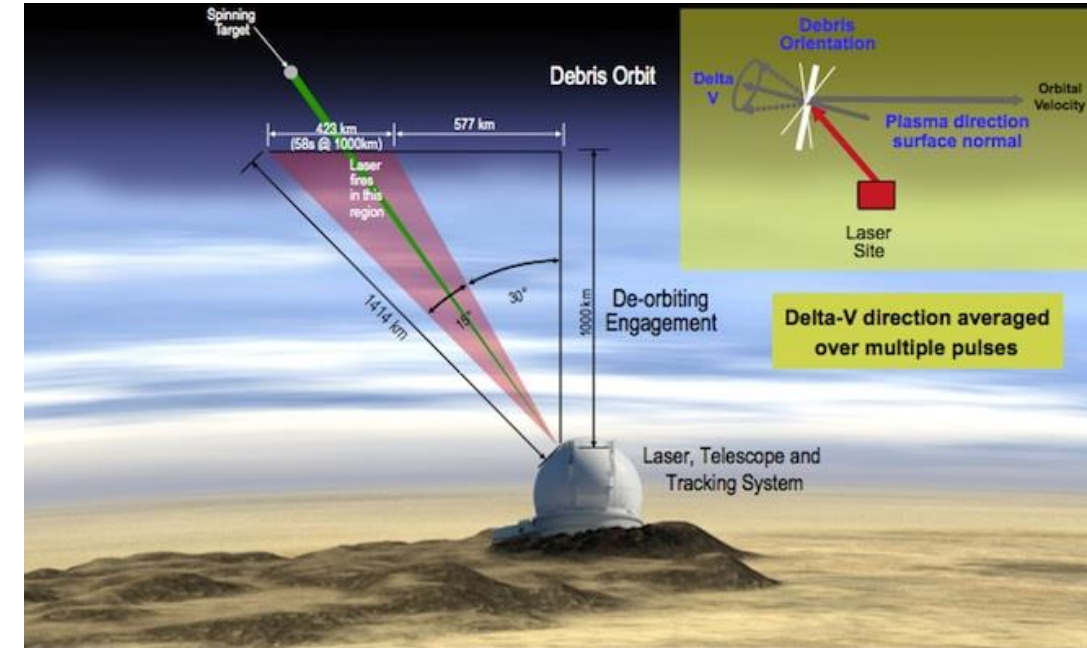
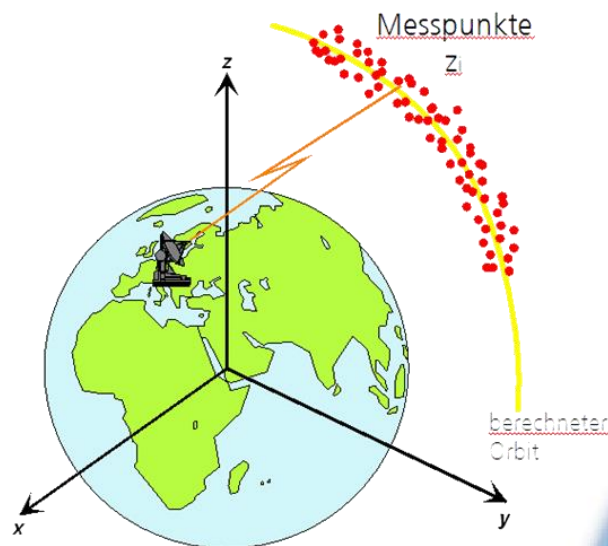


Simulierte Auswirkung des Aufpralls eines 12 mm großen Geschosses, mit der Geschwindigkeit von 7 km/s, auf eine 8 cm dicke Aluminiumplatte. Bild: ESA.



DLR-Software BACARDI

- Katalog aller Bahninformationen von bekannten Raumfahrtobjekten (Gruppe „Space Situational Awareness“)
- Welche mathematischen Methoden kommen zum Einsatz?
 - Bahnbestimmung für 1.000.000 Objekte
 - Propagation
 - Objektidentifikation
 - Kollisionsvorhersagen (Missionsunterstützung)
- „[Space Debris Viewer](#)“ ermöglicht eine 3D-Visualisierung von Objekten aus BACARDI.

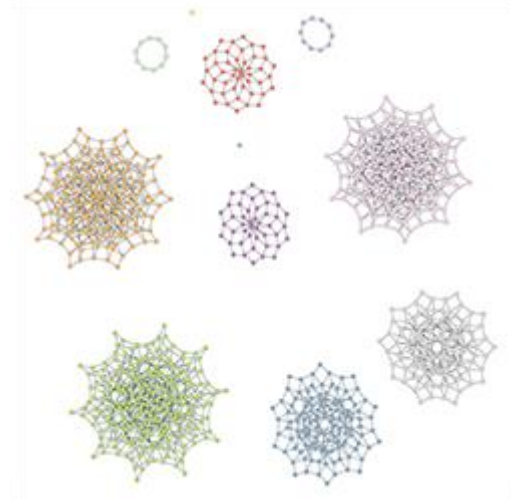


Tracking von Weltraummüll mit Hilfe von Lasern

Bildquelle <https://www.wired.com/2011/10/space-junk-laser/>

Exascale computing and Performance Engineering

ESSEX goes Oakforest-PACS



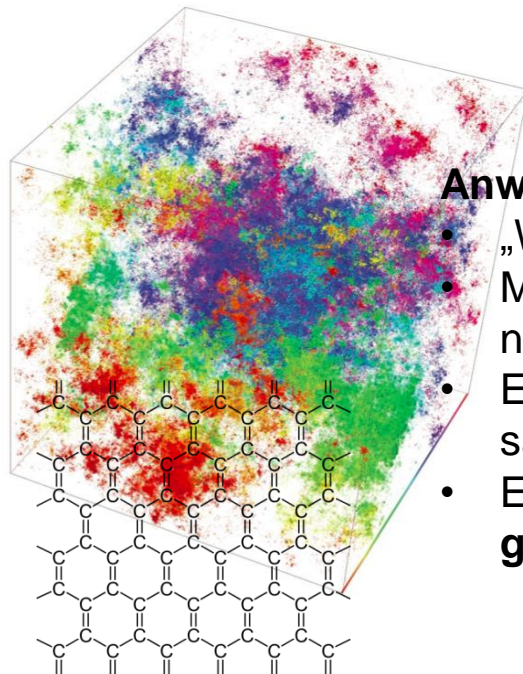
Graphvisualisierung der möglichen
Zustandsänderungen des Heisenberg
Spinkettenmodells



DFG Projekt ESSEX

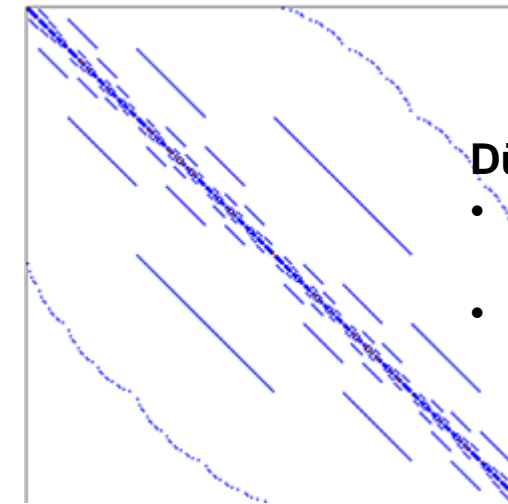
ESSEX Motivation: Requirements for Exascale

- Programmierkonzepte und numerische Algorithmen zur skalierbaren, effizienten und robusten Lösung dünnbesetzter Matrixprobleme auf Exascale Systemen.
- Breit anwendbare und skalierbare dünnbesetzten Eigenwertlösern mit hoher Hardware-Effizienz
- Zentrales Integrationsframework PHIST mit algorithmischen Grundbausteinen, linearen Lösern und der Jacobi-Davidson Methode bei.
- Beteiligung an GHOST (optimierte Kernel) und CRAFT (Fehlertoleranz) → RRZE, FAU-HPC



Anwendung aus der Physik:

- „Wundermaterial“ **Graphen**
- Modifikation des Kohlenstoffs mit 2D bienenwabenförmiger Struktur
- Einsatz in Speicherzellen, Superkondensatoren und Akkus
- Eigenschaften mit Hilfe der **Schrödinger-gleichung beschreibbar**



Dünnbesetzte Matrix

- Hamilton Operator einer Spinkette
- Eigenwerte interessant z.B. zur **Simulation von Quantencomputern**

Scalability on Oakforest-PACS

seit 6 / 2018 auf Platz Nummer 12 der



Cores:	556,104
Memory:	919,296 GB
Processor:	Intel Xeon Phi 7250 68C 1.4GHz (KNL)
Interconnect:	Intel Omni-Path
Linpack Performance (Rmax)	13.554 PFlop/s
Theoretical Peak (Rpeak)	24.913 PFlop/s
Nmax	9,938,880
HPCG [TFlop/s]	385.479



JCAHPC

Impression of the Oakforest-PACS supercomputer at the Japanese joint center for advanced HPC (JCAHPC).

Eigenwertberechnung mit Block-Jacobi-Davidson Verfahren

Ziel:

Finde Eigenpaare (λ_j, v_j) einer großen dünnbesetzten Matrix in einem gewissen Zielraum des Spektrums:

$$Av_j = \lambda_j v_j$$

- Projiziere das Problem auf einen geeigneten Unterraum
 - Löse das kleine EW Problem
 - Löse eine Korrekturgleichung
 - Orthogonalisiere zu allen vorherigen Suchrichtungen
 - Erweitere den Suchraum
-
- Blockvariante: Berechne Korrekturgleichung für n_b EW gleichzeitig
 - Beschränkung der globalen Synchronisierung durch Nutzen von Blockvektoren
 - Gleichzeitiges Lösen der linearen Gleichungssysteme in separaten Krylov-Räumen
 - Zusammenfassen von spMMVM und inneren Produkten
 - Zeilenweise Speichern aller Blockvektoren

Röhrig-Zöllner, M., Thies, J., Kreutzer, M., Alvermann, A., Pieper, A., Basermann, A., Hager, G., Wellein, G., Fehske, H. (2015). Increasing the Performance of the Jacobi--Davidson Method by Blocking. *SIAM Journal on Scientific Computing*, 37(6), C697-C722.



Benchmarks

- Fixed number of 250 Jacobi-Davidson iterations
- No additional preconditioning

Matrices

- Symmetric 7-point Laplace, 8.4M rows/node
- General 7-point, some PDE, 2.0M rows/node

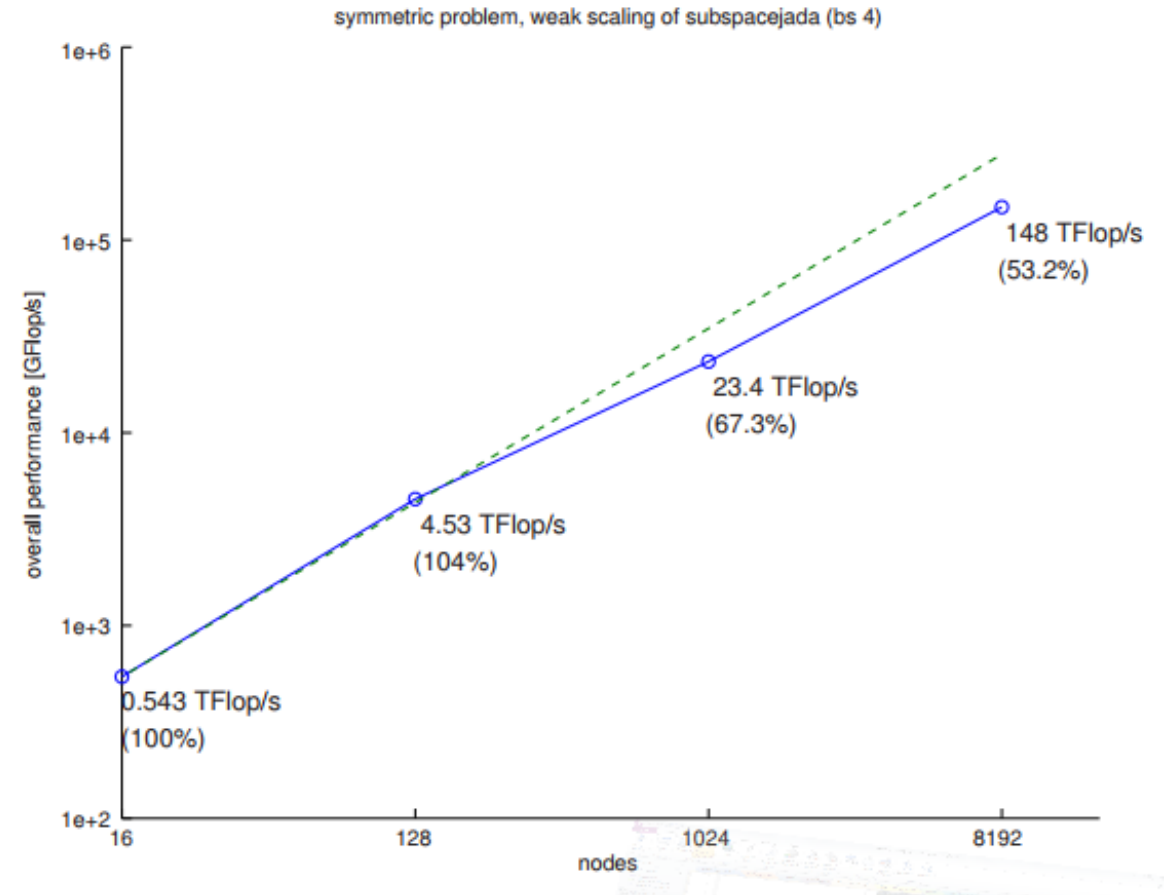
Solver parameters

- Krylov solver 10 iterations of MINRES (sym.)
- or GMRES+IMGS ortho (general)
- JD basis 16-40 vectors
- target eigenpairs near 0



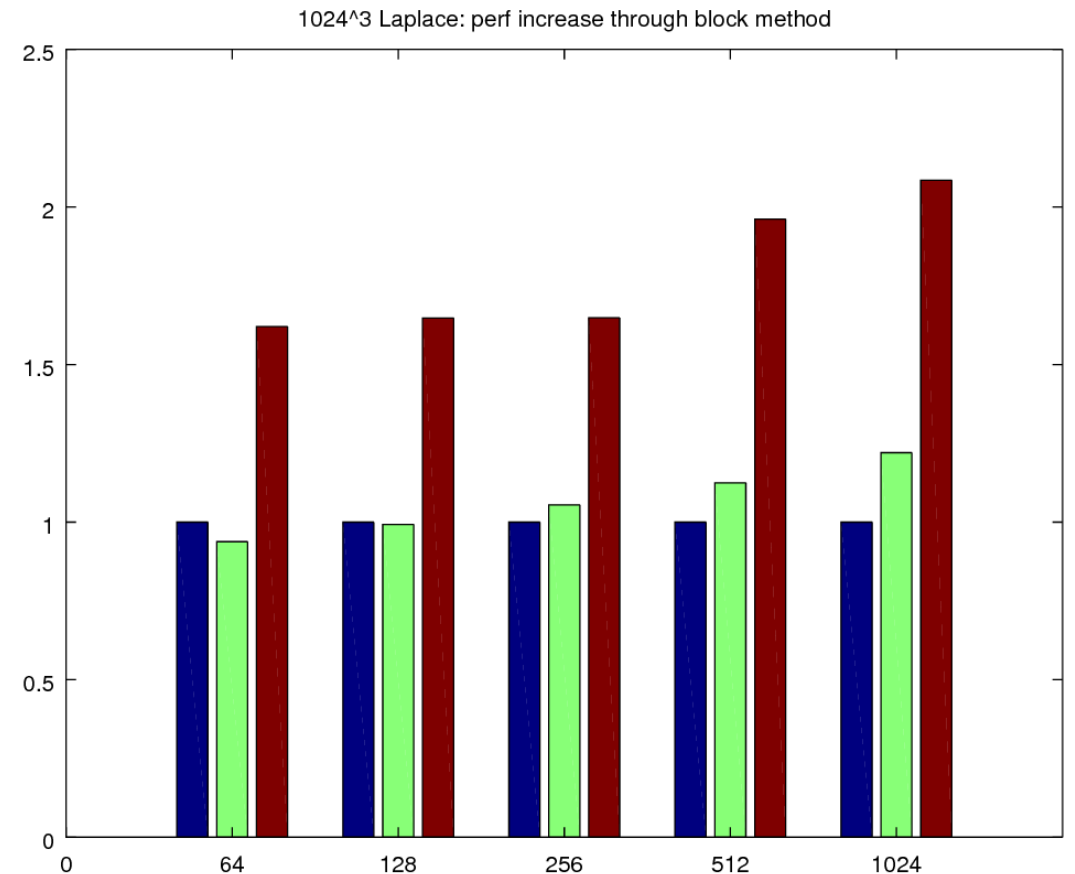
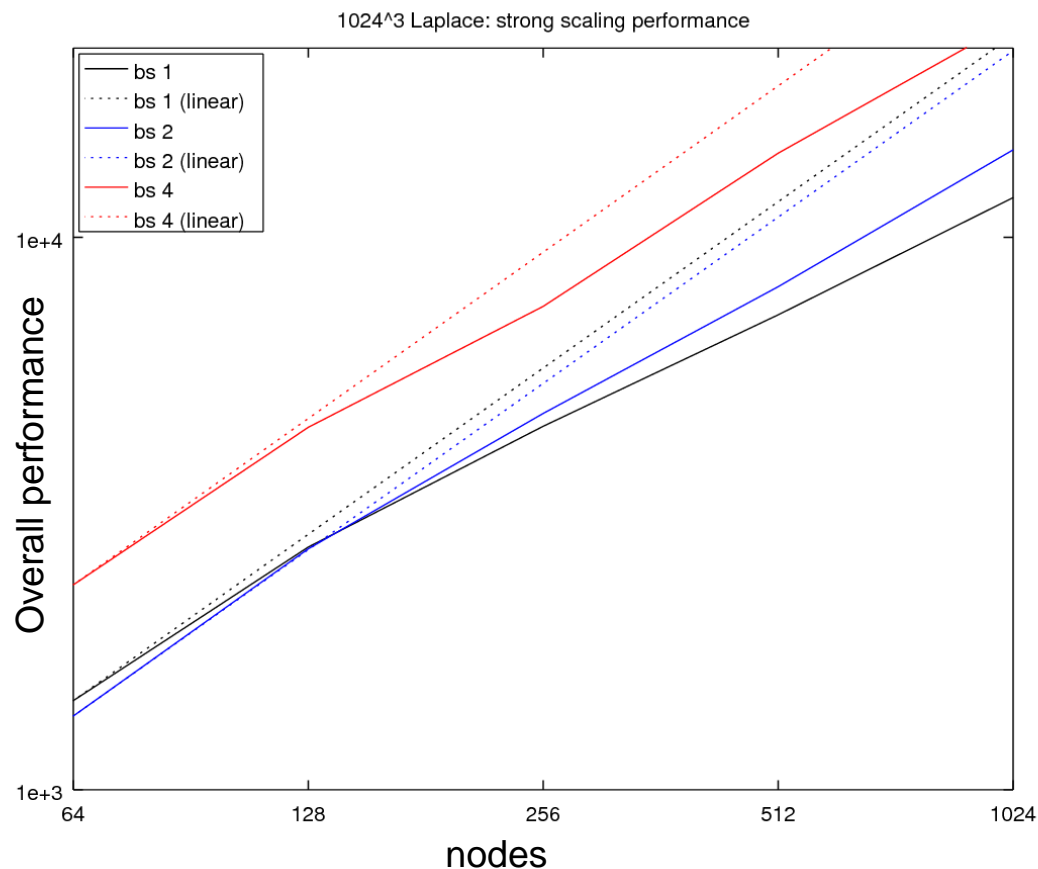
Weak scaling

- Up to 0.5M cores
- Percentage indicates the parallel efficiency compared to the first measurement (smallest node count).
- Symmetric PDE problem with the largest matrix size $N = 4\,0963$,
- The best performance was obtained with a block size of 4.



Strong scaling

- larger block size reduces number of Allreduce operations.



- corresponding 'block speedup' over the bs=1 case.
- The KNL doesn't seem to 'like' block size 2 very much (in contrast to XeonCPUs).
- Maybe the bandwidth can't be saturated with SSE?



How to ensure the quality of the ESSEX software: Basics

- **Git** for distributed software development



- **Merge-request workflow** for code review; changes only in branches

- [Visualization](#) of git repository development

```
[*****] Running 1 test from 1 test case.  
[-----] Global test environment set-up.  
[-----] 1 test from AddTest  
[ RUN ] AddTest.TwoAndTwo  
test2.cc:6: Failure  
Expected: Add(2, 2)  
Which is: 4  
To be equal to: 5  
[ FAILED ] AddTest.TwoAndTwo (0 ms)  
[-----] 1 test from AddTest (0 ms total)  
  
[-----] Global test environment tear-down  
[=====] 1 test from 1 test case ran. (1 ms total)  
[ PASSED ] 0 tests.  
[ FAILED ] 1 test, listed below:  
[ FAILED ] AddTest.TwoAndTwo  
  
1 FAILED TEST
```

- Own MPI extension for **Google Test**

- Realization of **continuous-integration** with Jenkins server



Paralleles Maschinelles Lernen, ein Ausblick

A large, high-resolution image of the Earth from space occupies the bottom right portion of the slide. It shows a curved horizon of the planet with a deep blue atmosphere. The surface features a mix of green landmasses, white clouds, and blue oceans. The text "Knowledge for Tomorrow" is overlaid on this image in a white, serif font.

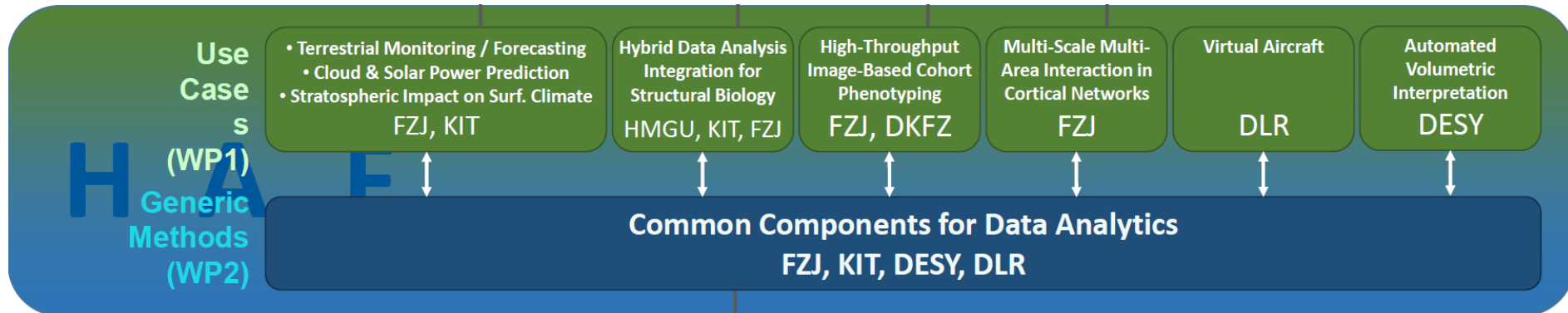
Knowledge for Tomorrow

HELMHOLTZ Analytics Framework

- Gemeinschaftsprojekt der 6 Helmholtz-Zentren

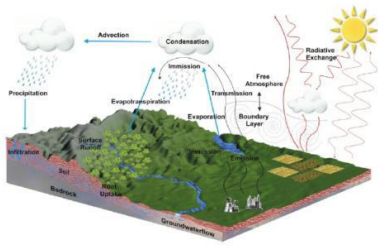


- **Ziel:** Entwicklung gemeinsamer Werkzeuge und Methoden zur Datenanalyse
- Beitrag von SC-HPC liegt hauptsächlich im WP2

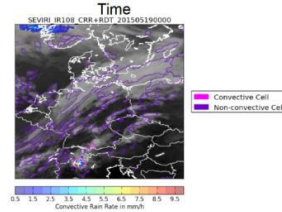


HAF Use Cases

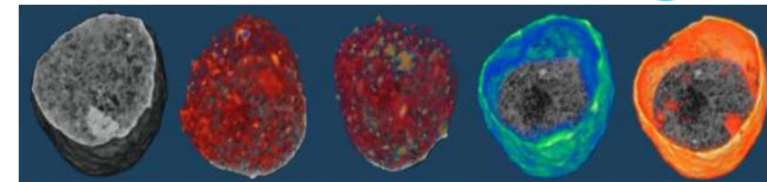
Earth System Modelling



SEVIRI Satellite Images – Near Real Time



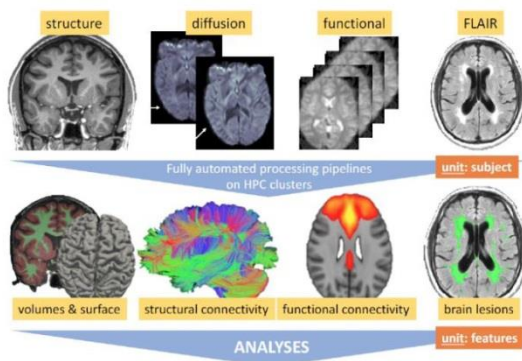
Research with Photons



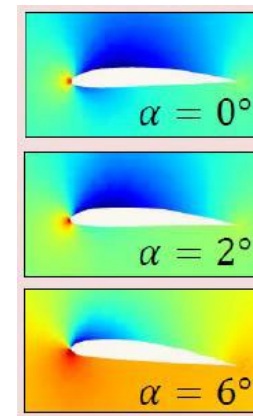
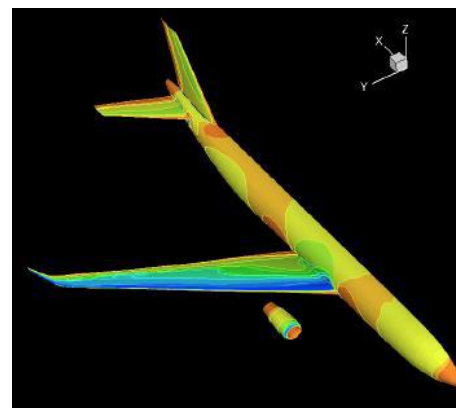
Neuroscience



DEUTSCHES
KREBSFORSCHUNGSZENTRUM
IN DER HELMHOLTZ-GEMEINSCHAFT



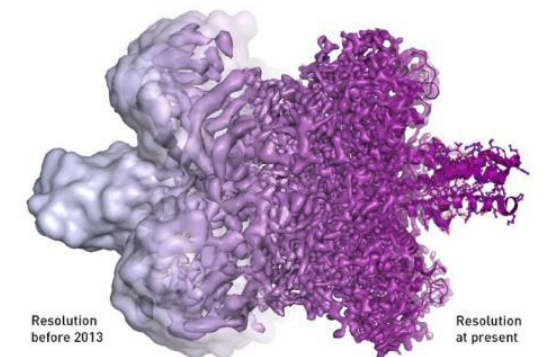
Aeronautics and Aerodynamics



Structural Biology



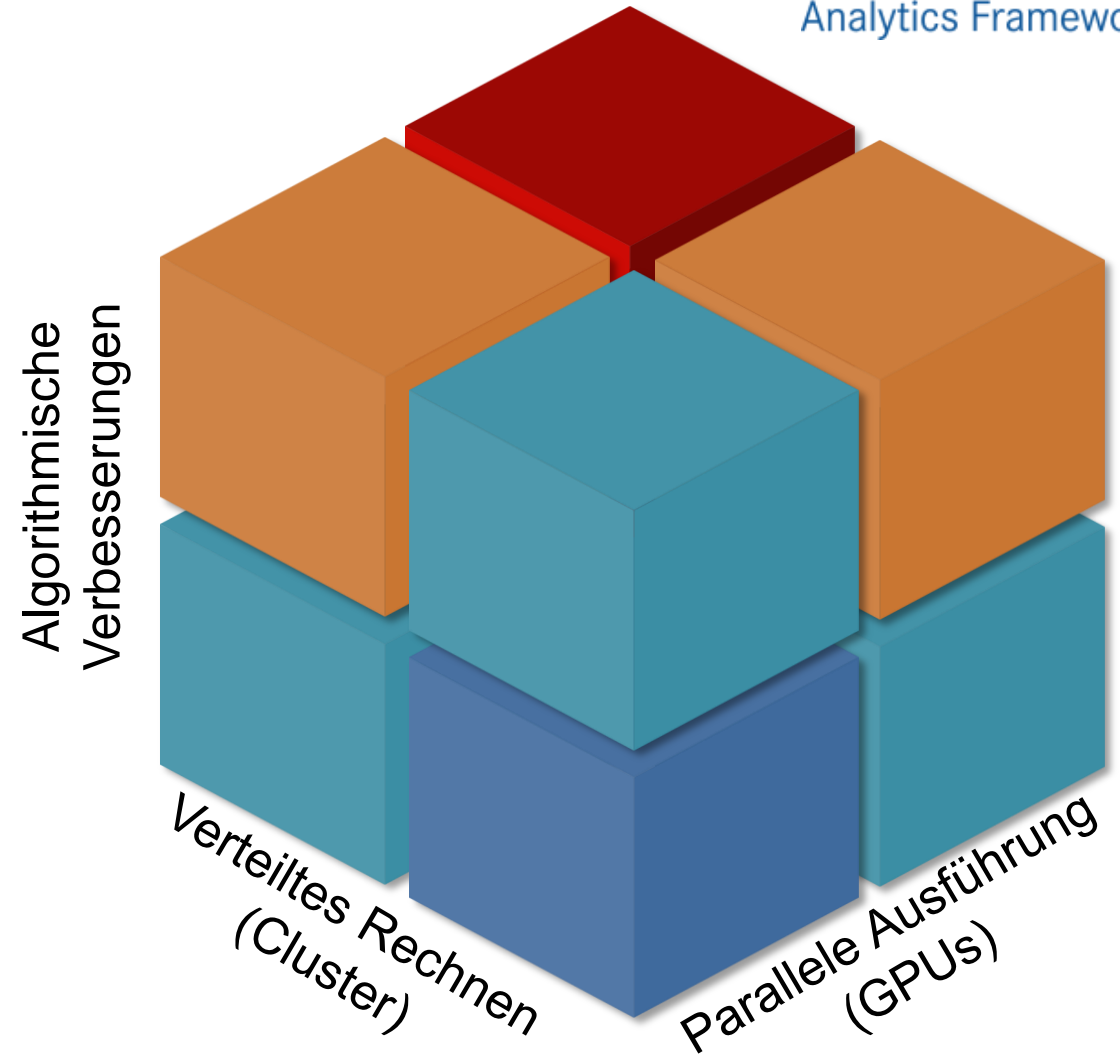
HelmholtzZentrum münchen
German Research Center for Environmental Health



Ziele von SC-HPC

- Maschinelles Lernen und High-Performance Computing verbinden
- Optimierungsalgorithmen für Deep-Learning verbessern
 - Einfluss der Gradientenberechnung auf Optimierer
- Algorithmen für extrem große Datenmengen

HELMHOLTZ
Analytics Framework



Was ist HeAT?

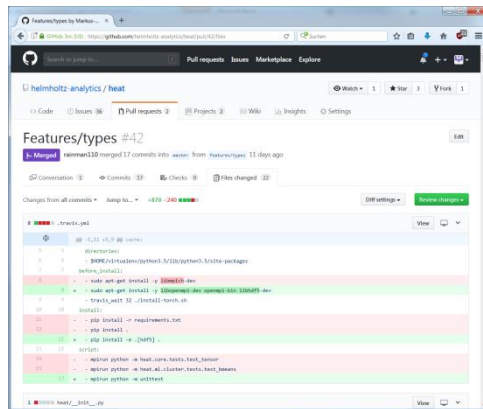
- **HeAT** = **He**lmholtz **A**nalytics **T**oolkit → The **hot** new machine learning / data analytics framework to come.
- Developed in the open:
Available on <https://github.com/helmholtz-analytics>
and <https://pypi.org/project/heat>
- Liberally licensed: MIT
- Zielsetzung:
 - Framework für generelle Machine Learning Aufgaben
 - Effiziente Ausnutzung von **verteilten** Rechnerarchitekturen (CPUs & GPUs)
 - numpy-artiges Interface zur einfachen Bedienbarkeit
- Basiert auf pytorch

HELMHOLTZ
Analytics Framework

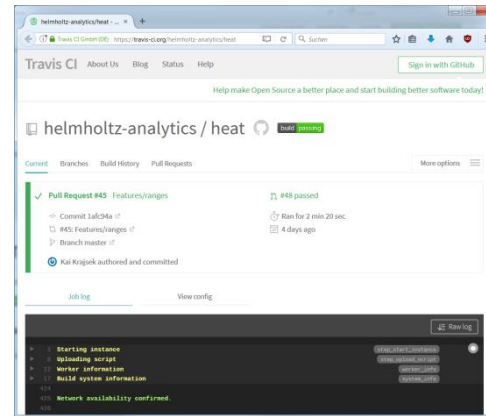


Transparent development process

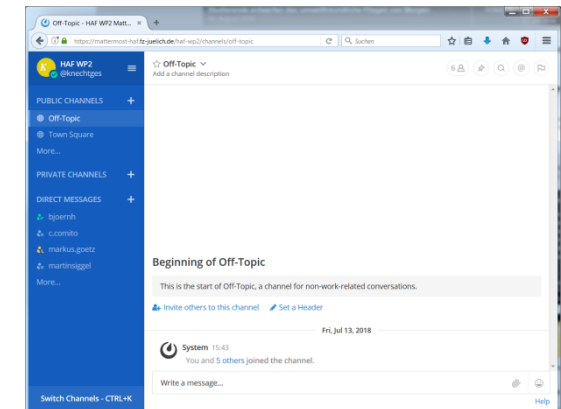
Github for code review,
issue tracking,
sprint planning



Travis for continuous integration



Mattermost for discussions



Join us there!

Many thanks for your attention!

Questions?

Dr. Margrit Klitz

German Aerospace Center (DLR)

Simulation and Software Technology

Department High Performance Computing

Margrit.Klitz@dlr.de

<http://www.DLR.de/sc>



We are hiring: <http://www.dlr.de/jobs/>

